--12. The method of claim 11, further comprising testing the gate-level designs for conformance with gate-level design requirements of the individual sub-modules prior to integrating the gate-level designs to form the top-level design.--

--13. The method of claim 12, wherein testing the gate-level designs include performing static timing analysis on the individual sub-modules for conformance with timing requirements for the individual sub-blocks.--

A1 cont

--14. The method of claim 13, wherein the gate-level netlists are generated for the sub-modules only if the timing requirements for the individual sub-modules are met.--

--15. The method of claim 14, wherein the step of synthesizing is re-performed and the gate-level designs are re-tested in an iterative manner for verifying conformance of the gate-level designs with the timing requirements of the individual sub-modules.--

--16. The method of claim 15, wherein the step of synthesizing gate-level designs is further based on wire loads and input/output loads/drivers.--

--17. The method of claim 16, wherein the step of verifying conformance of the gate-level designs includes performing dynamic simulations on the gate-level designs.--

REMARKS

Claims 1-9 are pending. The drawings, Abstract, Specification and claims 5 and 7 are amended to obviate minor informalities and not amended to overcome the applied art. Further, claim 4 is canceled and 1 is amended to incorporated the subject matter of claim 4. Claims 10-17 are added. No new matter is added.

The Office Action raises various objections to the drawings due to minor informalities. By this Amendment, the drawings are replaced to obviate the objections and the Detailed Description is amended to reflect the changes in the drawings. Accordingly, withdrawal of the objections is respectfully requested.

The Office Action rejects claims 1-4 and 6-9 under 35 U.S.C. §102(e) over Dupenloup (U.S. Patent No. 6,296,636), and further rejects claim 5 under 35 U.S.C. §103(a) over Dupenloup. These rejections are respectfully traversed with respect to claims 1-3 and 5-9, and further traversed to the extent that they apply to claims 10-17.

Claims 1-3 and 5-9 Define Patentable Subject Matter

In particular, Applicants assert that Dupenloup does not teach or suggest a method of synthesizing a register transfer level based design of a system including at least the steps of determining a plurality of sub-modules of a top level system, determining individual time budgets for each sub-module based on timing requirements of the top-level system, synthesizing gate-level designs of the sub-modules based on the determined time budgets for the individual sub-modules, and testing the gate-level designs for conformance with gate-level design requirements of the individual sub-modules, then integrating the gate-level designs of the individual sub-modules to form a top level design, as recited in independent claim 1.

Dupenloup discloses a method for generating scripts to synthesize RTL code. See, Abstract, for example. The Dupenloup method uses a complementary approach of top-down characterization and bottom-up synthesis steps. As shown in figure 19, the top-down step is used to provide constraints, time budgets and other information to be met by each module of a larger design. See also, col. 43, lines 16-19, for example. Once the constraints are formed, the various modules are synthesized, then integrated along a design hierarchy. See, col. 41, lines 1-12, for example. Dupenloup does not teach or suggest testing the gate-level designs for conformance with gate-level design requirements of individual sub-modules, then integrating the gate-level designs of the individual sub-modules.

To the contrary, while Dupenloup discloses synthesizing various modules based on time-constraints, nowhere does Dupenloup disclose, suggest or even appreciate any form of individual sub-module testing. See, col. 43, lines 8-26, for example. Although the Office Action asserts that Dupenloup discloses "testing the gate-level designs . . . prior to integrating the gate-level designs to form the top-level design (see, page 5, paragraph 7.4 of the Office Action), nowhere do the cited passages disclose or suggest performing any testing whatsoever on individual sub-modules. In fact, nowhere in Dupenloup does the term "test" occur in conjunction with individual sub-modules. Furthermore, nowhere in either of figures 14 or 19 (cited by the Office Action) show any block or step that suggests any form of sub-module testing. Thus, Dupenloup does not teach or suggest each and every limitation of independent claim 1.

Therefore, independent claim 1 defines patentable subject matter. Claims 2-3 and 5-9 define patentable subject matter by virtue of their dependency as well as for the additional features they recite. Accordingly, withdrawal of the rejections of claims 1-3 and 5-9 is respectfully requested.

Claims 10-17 Define Patentable Subject Matter

Applicants assert that Dupenloup does not teach or suggest a method of synthesizing a register transfer level based design of a system including at least the steps of determining a plurality of sub-modules of a top level system, determining individual time budgets for each sub-module based on timing requirements of the top-level system, . . . generating gate-level netlists for the gate-level designs of each of the sub-modules, and integrating the gate-level designs of the individual sub-modules, as recited in independent claim 10.

Dupenloup teaches the methods and systems as described above. Dupenloup does not teach or suggest at least generating gate-level netlists for the gate-level designs of each of a number of sub-modules.

To the contrary, while the Office Action uses the term "netlist" extensively, nowhere does Dupenloup describe generating a netlist from a sub-module. While Dupenloup does state that a "dump script technique used to extract the generic netlist from a Design Compiler has several benefits" (see col. 21, lines 29-33), nowhere does Dupenloup suggest generating a netlist from a sub-module via a "Design Compiler" or otherwise. Further, the specific figures and passages cited by the Office Action do not disclose or suggest generating gate-level netlists for the gate-level designs, but merely speak to high-level processes without referring to netlists whatsoever. In fact, Fig. 14 and its associated text seem to suggest a bottom-up approach where individual modules are created, frozen and then piecemeal integrated entirely within a design compiler without generating any intermediate netlists. Thus, Dupenloup does not teach or suggest each and every limitation of independent claim 10.

Therefore, independent claim 10 defines patentable subject matter. Claims 11-17 define patentable subject matter by virtue of their dependency as well as for the additional features they recite. Accordingly, withdrawal of the rejections of claims 10-17 is respectfully requested.

Conclusion

Applicants believe that this application is in condition for allowance and Applicants request that the Examiner give the application favorable consideration and permit it to issue as a patent. However, if the Examiner believes that the Application can be put in even better condition for allowance, the Examiner is invited to contact Applicants' representative listed below.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 500417 and please credit any excess fees to such deposit account.

Respectfully submitted,

MCDERMOTT, WILL & EMERY

B. Y. Mathis
Registration No. 44,907

600 13th Street, N.W.
Washington, D.C. 20005
(202) 756-8600 BYM/AJS:kap
Date: March 7, 2002
Facsimile: (202) 756-8000

Attachments:
Appendix With Markings to Show Changes Made
Request for Approval of Drawing Amendment with replacement Figures 1-4

APPENDIX WITH MARKINGS TO SHOW CHANGES MADE

IN THE ABSTRACT:

The Abstract has been amended as follows:

A method for synthesizing a registered transfer level (RTL) based design employs a bottom-up approach to generate a final top-level design. The top-level design is divided into a plurality of sub-modules. Each of the sub-modules is then independently synthesized using an RTL based design approach and independently adapted to conform to [the] timing requirements produced for each of the sub-modules [produced through time-budgeting] using time budgets that are based on the top-level timing requirements. Once the sub-modules are synthesized and pass individual timing requirements specific for those sub-modules, the sub-modules are integrated to form a top-level design. The top-level design may then be verified for timing requirements and other formal requirements.

IN THE SPECIFICATION:

The specification has been amended as follows:

On pages 3-4, please replace paragraphs the last three paragraphs of page 3 as follows:

As depicted in Figure 2, the timing requirements of the top-level design are used in the present invention to form a time-budget for the system, as depicted by block 26. Time-budgeting is a process of defining all of the timing requirements of input and output signals of each sub-module design to achieve functional operability as well as speed of operation of the top-level design. Hence, the time-budget 26 process provides the timing for the different sub-modules A, B, C. This is depicted in the synthesis of Figure 2 by blocks [28] 28A, 28B and 28C. The timing requirements of sub-modules A, B, C, provided in [block 28] blocks 28A, 28B and 28C, are used in the synthesis process of each of the individual sub-modules.

The top-level RTL is broken down into 3 logical RTL blocks, sub-module A RTL, sub-module B RTL and sub-module C RTL. These are represented by blocks [30] 30A, 30B and 30C in Figure 2. With the input of the timing requirements for the individual sub-module, and the sub-module RTL, broken down from the top-level RTL, a synthesis of each sub-module as a stand-alone design is performed as depicted in blocks [32] 32A, 32B and 32C. The synthesis process of an individual sub-module will be described later with respect to Figure 3. The synthesis of each sub-module design is performed independently, and generates a stand-alone design. Such a synthesis is performed in accordance with conventional RTL and synthesis techniques for the individual sub-module designs.

Each sub-module A, B, C undergoes an iterative process in order to meet the timing requirements of its input and output signals. Once a static timing analysis performed on each sub-module design verifies the timing requirements, a sub-module netlist is produced, as depicted in blocks [34] 34A, 34B and 34C. The netlist is a list of components and connections for the sub-modules.

On page 4, please replace paragraph 3 as follows:

Figure 3 depicts a typical flow of the synthesis of an RTL-based design for a sub-module, as depicted by [block 32] blocks 32A, 32B and 32C in Figure 2. This synthesis of an RTL-based design may be used for top-level designs or for individual sub-module designs. The synthesis finds particular application in the present invention for synthesizing sub-modules, which are then integratable into a top-level design, rather than using the synthesis to directly synthesize the top-level design.

On page 5, please replace paragraphs 2 and 3 as follows:

The procedures described above are summarized in the flow chart of Figure 4 in which the top-level timing requirements are provided in step 60. The time-budget of each of the sub-modules determines the timing requirements for each of the sub-modules, as provided in step 62. Each of the sub-modules undergoes independent gate-level synthesis in steps [64] 64A-C. The gate-level synthesis for the sub-module design is based on the timing requirements, wire load modules, and I/O signal loadings, for example. Verification of the performance of the gate-level design of the sub-modules are performed in [step 65] steps 65A-C. These include static timing analysis, dynamic simulations and other formal verifications. It is then determined in steps [66] 66A-C. for each of the sub-modules whether the timing requirements for the sub-modules are met. If they are not met, the synthesis process for the individual sub-module or sub-modules are performed until the timing requirements are met and verifications passed.

Once an individual sub-module has passed the timing requirements and verifications, as determined in [step 66] steps 66A-C, a gate-level netlist is provided in [step 68] steps 68A-C to be integrated in step 70 with the other gate-level netlists of the other sub-modules to form a top-level design netlist. The integrated top-level design netlist is then tested in step 71 in the same manner as each of the individual sub-modules in [step 65] steps 65A-C. It is determined in step 72 whether the integrated top-level netlist and design satisfies all of the top-level timing requirements and other verifications performed on the top-level final design. If it does not, the process returns to step 62 or steps [64] 64A-C to re-perform the synthesis of the sub-modules.

IN THE CLAIMS:

Claim 4 has been cancelled without prejudice or disclaimer.

Claims 1, 5 and 7 have been amended as follows:

1. (Amended) A method of synthesizing a register transfer level (RTL) based design of a system comprising the steps of:

determining [sub-module] a plurality of sub-modules of a top level system;

determining individual time budgets for each sub-module based on timing requirements of the top-level system;

synthesizing gate-level designs of the [sub-module] sub-modules based on the determined time budgets for the individual sub-modules;

testing the gate-level designs for conformance with gate-level design requirements of the individual sub-modules, then integrating the gate-level designs of the individual sub-modules to form a top level design;

testing the top-level design for conformance with top-level design requirements; and

generating a top-level netlist when the top-level design conforms to the top-level design requirements.

5. (Amended) The method of claim [4] 1, wherein testing the gate-level designs [include] includes performing static timing analysis on the individual sub-modules for conformance with timing requirements for the individual sub-blocks.

7. (Amended) The method of claim 6, wherein the step of synthesizing is re-performed and the gate-level designs are re-tested in an iterative manner [for verifying] to verify

conformance of the gate-level designs with the timing requirements of the individual sub-modules.

New claims 10-17 have been added as follows:

--10. A method of synthesizing a register transfer level (RTL) based design of a system comprising the steps of:

determining sub-module of a top level system;

determining individual time budgets for each sub-module based on timing requirements of the top-level system

synthesizing gate-level designs of the sub-module based on the determined time budgets for the individual sub-modules;

integrating the gate-level designs of the individual sub-modules to form a top level design;

testing the top-level design for conformance with top-level design requirements;

generating gate-level netlists for the gate-level designs of each of the sub-modules; and

generating a top-level netlist when the top-level design conforms to the top-level design requirements.--

--11. The method of claim 10, wherein the step of integrating the gate-level designs includes integrating the gate-level netlists of the sub-modules.--

--12. The method of claim 11, further comprising testing the gate-level designs for conformance with gate-level design requirements of the individual sub-modules prior to integrating the gate-level designs to form the top-level design.--

--13. The method of claim 12, wherein testing the gate-level designs include performing static timing analysis on the individual sub-modules for conformance with timing requirements for the individual sub-blocks.--

--14. The method of claim 13, wherein the gate-level netlists are generated for the sub-modules only if the timing requirements for the individual sub-modules are met.--

--15. The method of claim 14, wherein the step of synthesizing is re-performed and the gate-level designs are re-tested in an iterative manner for verifying conformance of the gate-level designs with the timing requirements of the individual sub-modules.--

--16. The method of claim 15, wherein the step of synthesizing gate-level designs is further based on wire loads and input/output loads/drivers.--

--17. The method of claim 16, wherein the step of verifying conformance of the gate-level designs includes performing dynamic simulations on the gate-level designs.--